CONCURRENT ENGINEERING: Research and Applications

CAD/CAPP Integration using Feature Ontology

Christel Dartigues,[1,*] Parisa Ghodous,[2] Michael Gruninger,[3] Denis Pallez[2] and Ram Sriram[3]

[1]I3S UNSA-CNRS – 2000, route des lucioles, Les Algorithmes – bât. Euclide B
BP.121, 06903 Sophia Antipolis – Cedex – France
[2]LIRIS, University Claude Bernard of Lyon, 43, Boulevard du 11 novembre 1918, 69622 Villeurbanne Cedex, France
[3]NIST, 100 Bureau Drive, Gaithersburg, MD 20899-3460, USA

Abstract: In a collaborative computer-supported engineering environment, the interoperation of various applications will need a representation that goes beyond the current geometry-based representation, which is inadequate for capturing semantic information. The primary purpose of this study is to discuss a semantically based information exchange protocol that will facilitate seamless interoperability among current and next generation computer-aided design systems (CAD) and between CAD and other systems that use product data. An ontological approach is described to integrating computer-aided design (CAD) and computer-aided process planning (CAPP). Two commercial software applications are used to demonstrate the approach. This involves the development of a shared ontology and domain specific ontologies in the Knowledge Interchange Format (KIF) language. Domain specific ontologies – which are feature-based – are developed after a detailed analysis of the CAD and the CAPP software. Mapping between the domain ontologies and the shared ontology is achieved by several mapping rules. The approach is validated by using a variety of parts.

Key Words: CAD, CAPP, interoperability, ontologies, design, process planning, knowledge interchange format.

1. Introduction

The early part of this millennium has witnessed the emergence of an Internet-based engineering marketplace, where engineers, designers, and manufacturers from small and large companies are collaborating through the Internet to participate in various product development and marketing activities [1–4]. This will be further enhanced by the next generation manufacturing environment, which will consist of a network of engineering applications, where state of the art multimedia tools and techniques will enhance closer collaboration between geographically distributed applications, virtual reality tools will allow visualization and simulation in a synthetic environment, and information exchange standards will facilitate seamless interoperation of heterogeneous applications. The interoperation of various applications will need a representation that goes beyond the current geometry-based representation, which is inadequate for capturing semantic information. The primary purpose of this study is to discuss a semantically based information exchange protocol that will facilitate seamless interoperability among current and next generation computer-aided design systems (CAD) and between CAD and other systems that use product data. The focus will be on design/process planning integration during the later design stages. An approach using a neutral format based on feature ontology is presented. This work is divided into three main phases, that will be further explained in the rest of this study, as shown in Figure 1:

- The analysis of the two domains studied: detailed design and process planning,
- The creation of the ontology, and
- The definition and implementation of mapping rules.

In the next section, a brief overview of design/process planning integration is provided. This will be followed by a discussion of representative standards for interoperating design and process planning. The need for ontological approaches is presented followed by descriptions of ontologies in the design and process planning domains and a common ontology. Rules for mapping from and to the common ontology are described. Finally, the approach is illustrated with an example.

2. Design/Process Planning Integration: An Overview

Engineering a product involves several stages with considerable iterations [5]. In this study, an important

*Author to whom correspondence should be addressed.
E-mail: dartigue@unice.fr
Figures 1, 3, 10, 11 and 14 appear in color online: http://cer.sagepub.com

1063-293X/07/02 0237–13 $10.00/0 DOI: 10.1177/1063293X07079312

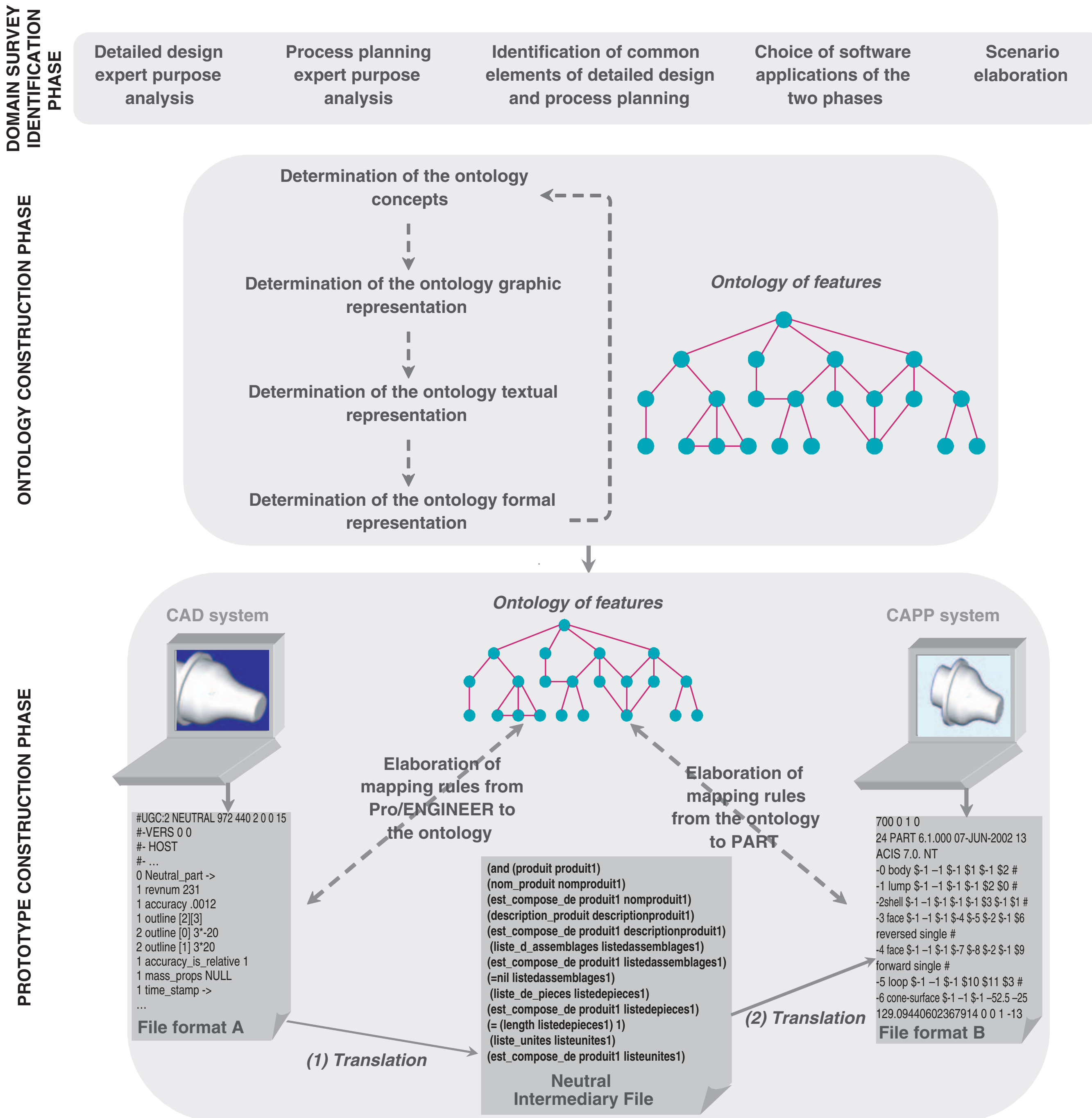

Figure 1. Global process for data exchange using ontologies.

aspect of the above cycle is focused: design and process-planning integration. It is believed that it is important to integrate design and process planning at various levels of abstraction, as errors made during early design stages could have a significant impact on the overall product quality and costs [5–8].

Engineering design involves mapping a specified function (or functional specifications) onto a (description of a) realizable physical structure – the design artifact. Over the past several decades considerable research has been done in developing various design product and process models [9]. The authors will not delve into a detailed description of the design process, much as they feel a need for the adequate representations for process knowledge. The reader is referred to [5] for a formal description of a design process model. At this stage the primary concern is on the product or artifact representation. For this, the NIST CORE product model presented in [10] is used.

Process planning is an intermediate phase between design and manufacture [11,12]. More precisely, it links these two decisive phases of product development [13]. It depends on choices made in design and determines precisely actions that will be achieved during manufacture. Different definitions have been given for process planning [7,14–16]. In this study, process planning is the phase that, from information generated during preliminary design (product geometry for instance), determines necessary operations and actions to transform a raw part in a finished or semi-finished part, the necessary human and material resources to manufacture the product, as well as the product development cost evaluation.

A wide variety of manufacturing processes are available for the actual artifact production. Here, the machining processes for part production are focused on, in particular material cutting processes. Figure 2 provides a representation of this process: the cutting tool comes against the surface, creating a chip that will be removed from the part.

The interactions between design and process planning occurs at various stages, from conceptual to detailed design/process planning as shown in Figure 3 [17]. Current interfaces between design and process planning are defined during the detailed design stage. This is primarily achieved through use of geometric features. However, there is considerable difference in the methods and terminology used: features are used to design a product (design by feature) [8,13,18,19] while in process planning they are extracted from the product (feature recognition or extraction) [13,18–21], and a consistent feature terminology does not exist for the two domains. These different viewpoints of designers and process planners on features makes data exchange a tedious task. Although features are considered differently in design and process planning, they represent a natural link between these two domains. Hence, features

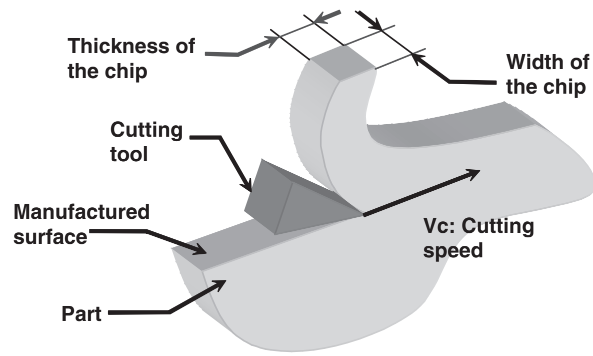

Figure 2. Representation of the material cutting process.

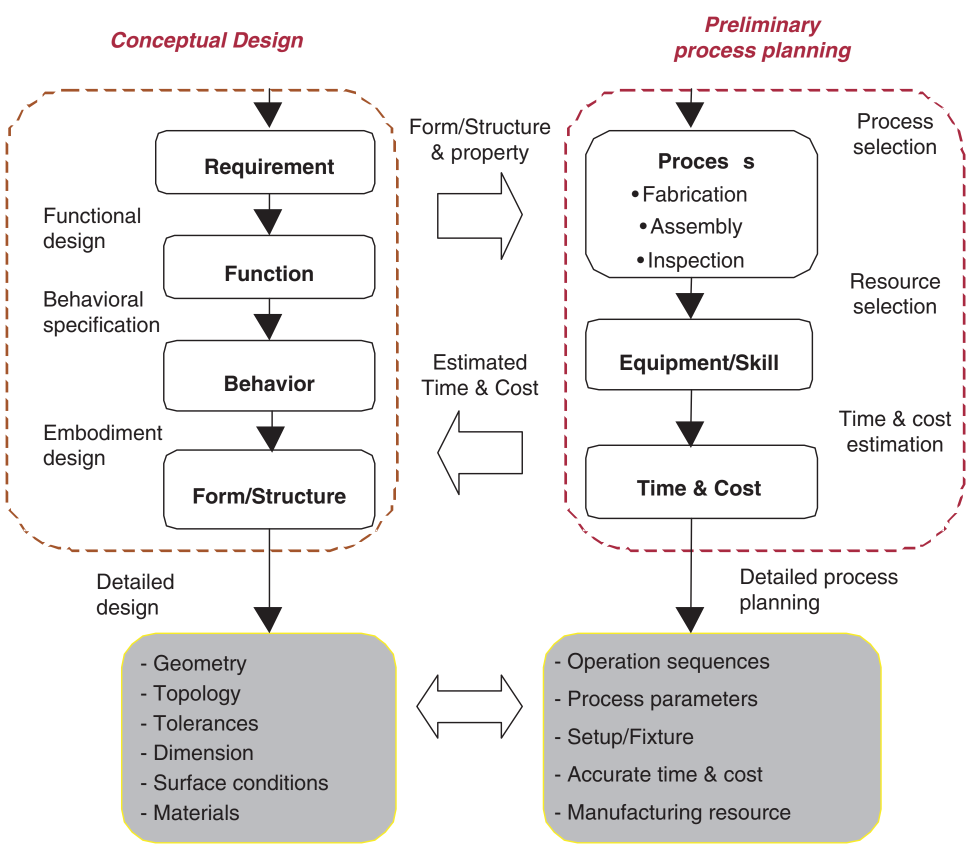

Figure 3. Design and process planning interfaces.

provide a valuable mechanism for information exchange. Next, the current standards in design and process planning interoperability are reviewed and extensions needed for feature-based interoperability are discussed.

3. Standards for Interoperability

The interoperability issue between CAD systems is illustrated by considering a potential information exchange scenario during the design of the Boeing 777. For Boeing to incorporate Rolls Royce engines into the design, the data format has to be converted from Computer Vision's CADDS (used by Rolls Royce) to Dassault's CATIA. Similarly, for Rolls Royce to understand changes made by Boeing engineers, the data need to be converted from CATIA to CADDS. Hence, this needs at least two translators. For three systems this grows to six translators and for n systems one needs $n(n-1)$ translators. Hence, there is a need to design, build, and maintain $n(n-1)$ translators. A solution to this problem is to use a neutral format and make all the CAD applications output this format. Doing so will reduce the number of translators to $2*n$, i.e., for each CAD system two translators are needed – one from the CAD system to the neutral format and the other from the neutral format to the CAD.

A standard of primary interest to design is ISO 10303, also known informally as Standard for the Exchange of Product model data (STEP) [3,22,23,34] and developed by the International Organization for Standardization Technical Committee 184/Subcommittee SC4 (ISO TC 184/SC4). Its intention is to enable the exchange of product model data between different modules of a product realization system, or the sharing of that data by different modules through the use of a common database [24]. The first parts of STEP to achieve International Standard status were published in 1994, but many other parts have since been published or are under development and will eventually be added to the standard. Recent updates (and other relevant details) can be found at the following websites: http://www.nist.gov/sc4, www.tc-184-sc4.org, and http://www.iso.ch/iso/en/ISOOnline.frontpage.

Considerable research has been performed on mapping CAD data onto process planning systems. However, this work has met with limited success, such as the one reported by [25]. One problem with the current standards is the lack of integration between CAD data output and process planning input. For example, the primary focus of STEP AP 203 is the interoperability between geometry centered CAD systems, while the focus of STEP AP 224 (mechanical product definition for process plans using machining features) has been on input to process planning systems with a primary focus on representation of machine features. The idea of features has been in vogue for some time and the literature is abound with definitions of features [15,19,26–32]. For example, Shah et al. suggest that features 'are primitive or low level designs with their attributes, qualifiers, and restrictions which affect functionality and/or manufacturability. Features can describe form (size and shape), precision (tolerances and finishing), or materials (type, grade, properties, and treatment), and vary with product and manufacturing process'.

To achieve truly collaborative design and engineering, exchange representations of both design and process information must support multiple levels of abstraction. To adequately achieve this one needs a more formal method for representing features, such as the ontological approach described in the next section. This approach has some similarities to the one presented in [33], but the overall methodology is different.

4. Ontological Approach to Interoperability

In all types of communication, the ability to share information is often hindered because the meaning of information can be drastically affected by the context in which it is viewed and interpreted. This is especially true in manufacturing because of the growing complexity of manufacturing information and the increasing need to exchange this information among various software applications. Different representations of the same information may be based on different assumptions about the world, and use differing concepts and terminology – and conversely, the same terms may be used in different contexts to mean different things. Often, the loosely defined natural-language definitions associated with the terms will be too ambiguous to make the differences evident, or will not provide enough information to resolve the differences.

To address these challenges, various groups within industry, academia, and government have been developing sharable and reusable models known as ontologies [3]. All ontologies consist of a vocabulary along with some specification of the meaning or semantics of the terminology within the vocabulary. In doing so, ontologies support interoperability by providing a common vocabulary with a shared semantics. Rather than develop point-to-point translators for every pair of applications, one simply needs to write one translator between the application's terminology and the common ontology. Similarly, ontologies support reusability by providing a shared understanding of generic concepts that span across multiple projects, tasks, and environments.

The various ontologies that have been developed can be distinguished by their degree of formality in the specification of meaning. With informal ontologies, the definitions are expressed loosely in natural language. Semi-formal ontologies, such as taxonomies, provide weak constraints for the interpretation of the terminology. Formal ontologies use languages based on mathematical logic. Informal and semi-formal ontologies can serve as a framework for shared understanding among people, but they are often insufficient to support interoperability, since any ambiguity can lead to inconsistent interpretations and hence hinder integration.

Another source of semantic heterogeneity lies in the languages used to represent the ontologies. There have been several efforts within academia and industry to develop common languages that can be used as the basis for ontologies to support semantic integration; the most expressive is the Common Logic project, which combines the Knowledge Interchange Format [34–36] and Conceptual Graphs (CG) [37] languages. Common Logic includes a core language that has the expressiveness of first-order logic; its syntax and semantics are those of traditional first-order logic. Some other languages have been based on Logic, such as PSL [38]. Most recently, this has been extended to include extensions that allow sorted formulae for the specification of class hierarchies, and the specification of the meta theory of KIF within the language itself.

The objective of this study consists in developing and implementing an approach for data exchange between designers and process planners. To realize this, it has been decided to develop a feature ontology. This ontology will represent all the common knowledge between designers and process planners as well as specific knowledge of both experts. This ontology will be used as depicted: a designer creates an artifact shape model using CAD software (such as Pro/Engineer); this model is then transformed, using mapping rules (see Section 8), into instances of the shared ontology. These instances of the shared ontology are then transformed, using other mapping rules, into a representation interpretable by CAPP software (such as Pro/Engineer). Features represent a common knowledge that will be the base of the shared ontology for data exchange. In the next parts of this study, the design specific parts will be presented, the process planning specific parts and the design and process planning common parts of the ontology. The study continues with the description of the mapping rules used to translate data and ends by an example.

5. Design Feature Ontology

The ultimate goal is to develop a comprehensive feature model that can be used through the entire design life cycle. However, for the prototype the NIST CPM's extensions were restricted to the information generated by commercial CAD systems. To identify these concepts, an extensive analysis was first performed to understand various designers' needs. This analysis phase involved:

- The extraction of designer know-how – which is implicit – in order to formalize designer's knowledge; and
- The analysis of different CAD softwares, such as Pro-Engineer and SolidWorks: these were used to create various parts in order to better understand the design process.

Based on this analysis it is concluded that the NIST CPM had most of the necessary classes to represent detailed design data. A few classes were added in order to increase the coverage to CAD software, such as: the datum coordinate system in which the artifact is defined, the dimensions associated to an artifact, the precision of the dimensions of an artifact, the different versions of an artifact, and the constraints associated to each feature. Figure 4 represents these concepts.

Different kinds of constraints were also defined as shown in Figure 5. The initial categories that were considered are position and orientation constraints, which can be further classified into attachment and geometric constraints. Attachment constraints specify how a feature instance is attached to the global model by coupling some of the feature faces with the pre-existing faces. Geometric constraints specify geometric relations such as parallelism of two faces or distance between two faces. Validity constraints correspond to another constraint category defined in the ontology. These validity constraints can be further classified into dimension constraints, algebraic constraints, boundary constraints, and feature interaction constraints.

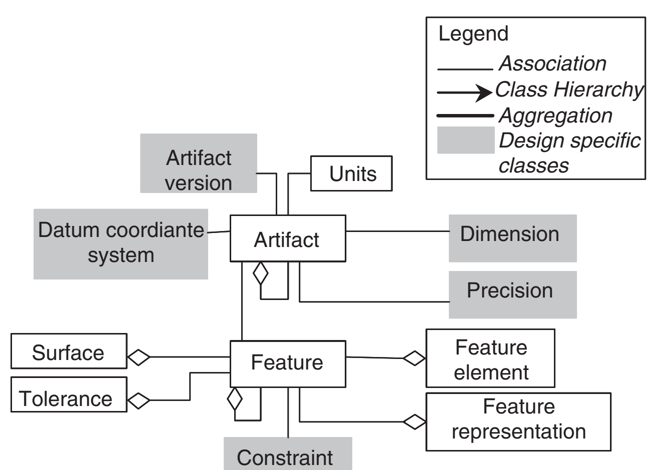

Figure 4. Design specific classes.

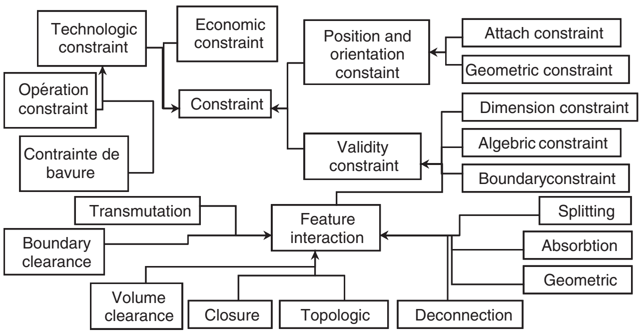

Figure 5. Constraint classification.

```
;A constraint is the super type of: technologic constraint,
;economic constraint, validity constraint and position and
;orientation constraint.
(forall (?a)
      (implies (constraint ?a)
             (or (technologic_constraint ?a)
                (economic_constraint ?a)
                (validity_constraint ?a)
                (position_orientation_constraint ?a))))

;A technologic constraint is the subtype of a constraint.
(forall (?a)
      (implies (technologic_constraint ?a)
             (constraint ?a)))

;A validity constraint is the subtype of a constraint.
(forall (?a)
      (implies (validity_constraint ?a)
             (constraint ?a)))
...
```

Figure 6. KIF statements for constraint classification.

The above extensions suffice to illustrate the approach. Additional classes will be needed for a wider coverage. KIF representations of a representative set are shown in Figure 6.

6. Process Planning Ontology

The feature ontology is also representative of the process planning viewpoint. A similar approach used for design was followed: process planners were asked to describe how they work, what kind of information they need, what are the different phases of their work, etc. CAPP software was also studied: PART. This analysis of process planning turned out to be a more difficult task than obtaining the design features. While designers have a consistent notion of what design is, process planners seem to be in less agreement on the terminology in their domain. Based on the discussions, it was decided to use the concepts presented in Figure 7.

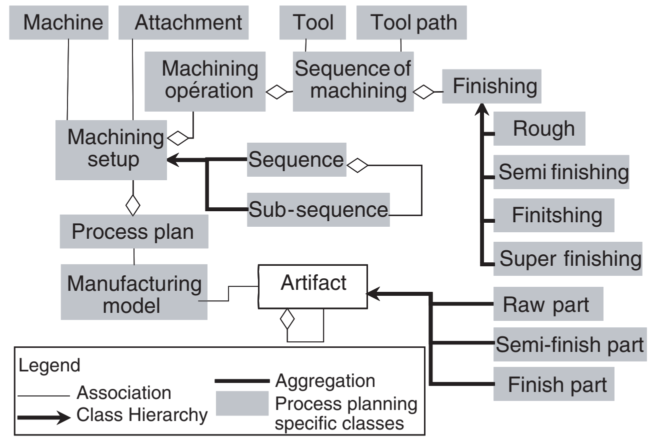

Figure 7. Process planning specific classes.

In this figure, an artifact is associated with a manufacturing model. This model is used to create a process plan. The input of this process plan is a raw part and the output is a semi-finished or finished part. A process plan identifies the machining operations that are necessary to manufacture an artifact. Hence, a process plan is composed of machining setups, which contains all the machining operations that are realized with the same machine and without changing the attachments. For each machining setup, there is a set of machining operations. Each machining operation is then realized with the same machine and attachments. Each machining operation is composed of a set of machining sequences, which corresponds to a transformation of a part that is achieved with the help of a material removal tool moving according to a tool path. Finally, a machining operation modifies a surface in accordance to a required finish: raw, semi-finish, finish, or super-finish.

7. Common Feature Ontology

The last part of the ontology corresponds to the common concepts between design and process planning and is composed of numerous classes and relationships. The ontology is based on the NIST Core Product Model (CPM). This model was used in order to take into account general concepts, initially present in this model, more specific concepts were added allowing feature representation. Figure 8 represents the main classes and relationships composing the Core Product Model and its extensions in this work, where the extensions are shown as darkened boxes (ideally, the NIST CPM should be a package in UML and the extensions should be in a separate package). The descriptions of key entities in the NIST CPM are as follows (taken from [10]).

An Artifact represents a distinct entity in a design, whether that entity is a component, product, subassembly, or assembly. The Artifact's attributes refer to the

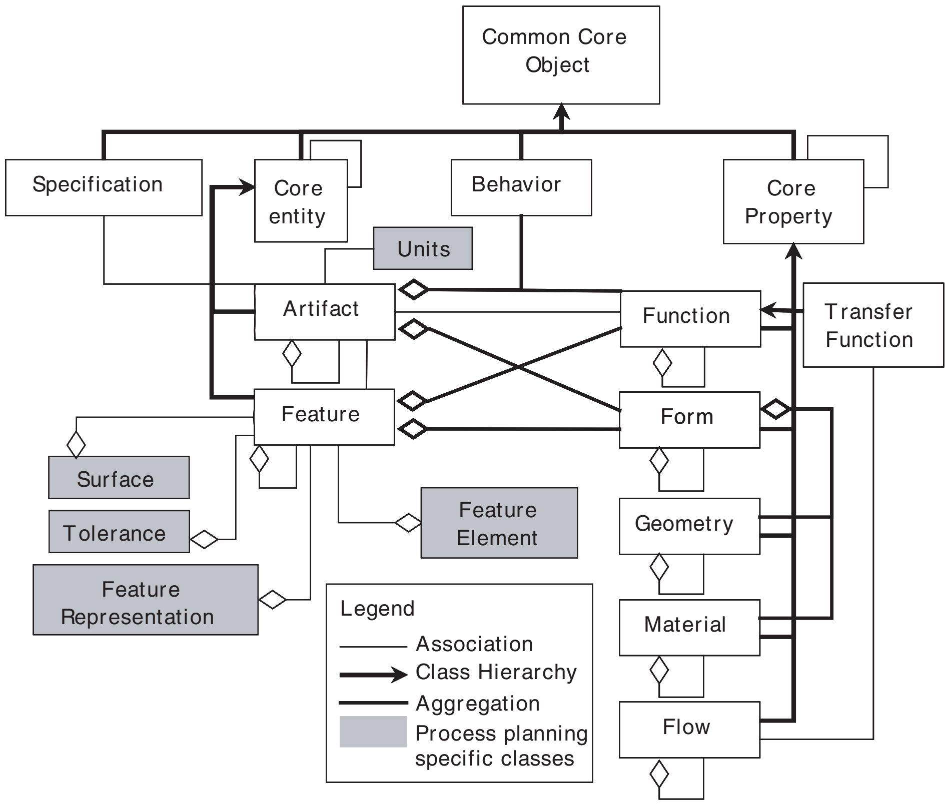

Figure 8. Main class diagram of the Core Product Model and extensions.

Specification responsible for the Artifact and the Form, Function, and Behavior comprising the Artifact. The Function represents what the Artifact is supposed to do. The Artifact satisfies the engineering requirements largely through its Functions. The Form of the Artifact can be viewed as the proposed design solution for the design problem specified by the Functions. More precisely, the physical characteristics of an Artifact are represented in terms of its geometry and material properties.

Another important class of the Core Product Model is the Feature. An Artifact is composed of a set of features, where a feature is a subset of the form of an object that has some function assigned to it. One can have several types of features: analysis features, design features, manufacturing features, interface or port features, etc. Compound features can be generated from primitive features. The notion of a feature is further elaborated in this work.

The NIST Core Product Model (CPM) [39] was modified by adding some concepts that are common to design and process planning, are both necessary for designers and process planners, and are considered in CAD and CAPP software.

The main objective is to find a common feature representation between design and process planning. To do so, NIST CPM was extended to address the following:

- The way each feature is represented, such as a B-Rep representation, a CSG representation, a swept representation, etc. (Feature Representation concept); and
- The elements composing each feature, such as a bottom side, an intermediary face, etc. (Feature Element concept).

A complete feature decomposition was also characterized which is based on the feature categories proposed in the part 48 of STEP [40]. Figure 9 illustrates this decomposition. Features are classified into: volume features, transition features, and pattern features. A more detailed description of this decomposition can be seen in [40,41].

8. Mapping Rules For Case Study

Once the feature ontology in various domains is defined, the next step is to define the mapping rules that will transform specific files onto instances of the common ontology. For the case study, the Pro-Engineer software was chosen, which is used by CAD

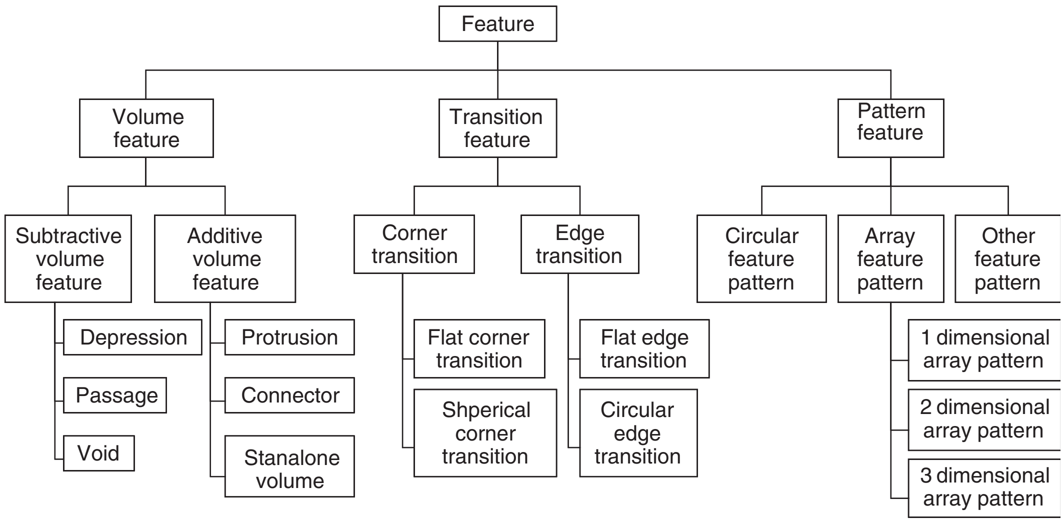

Figure 9. Feature decomposition.

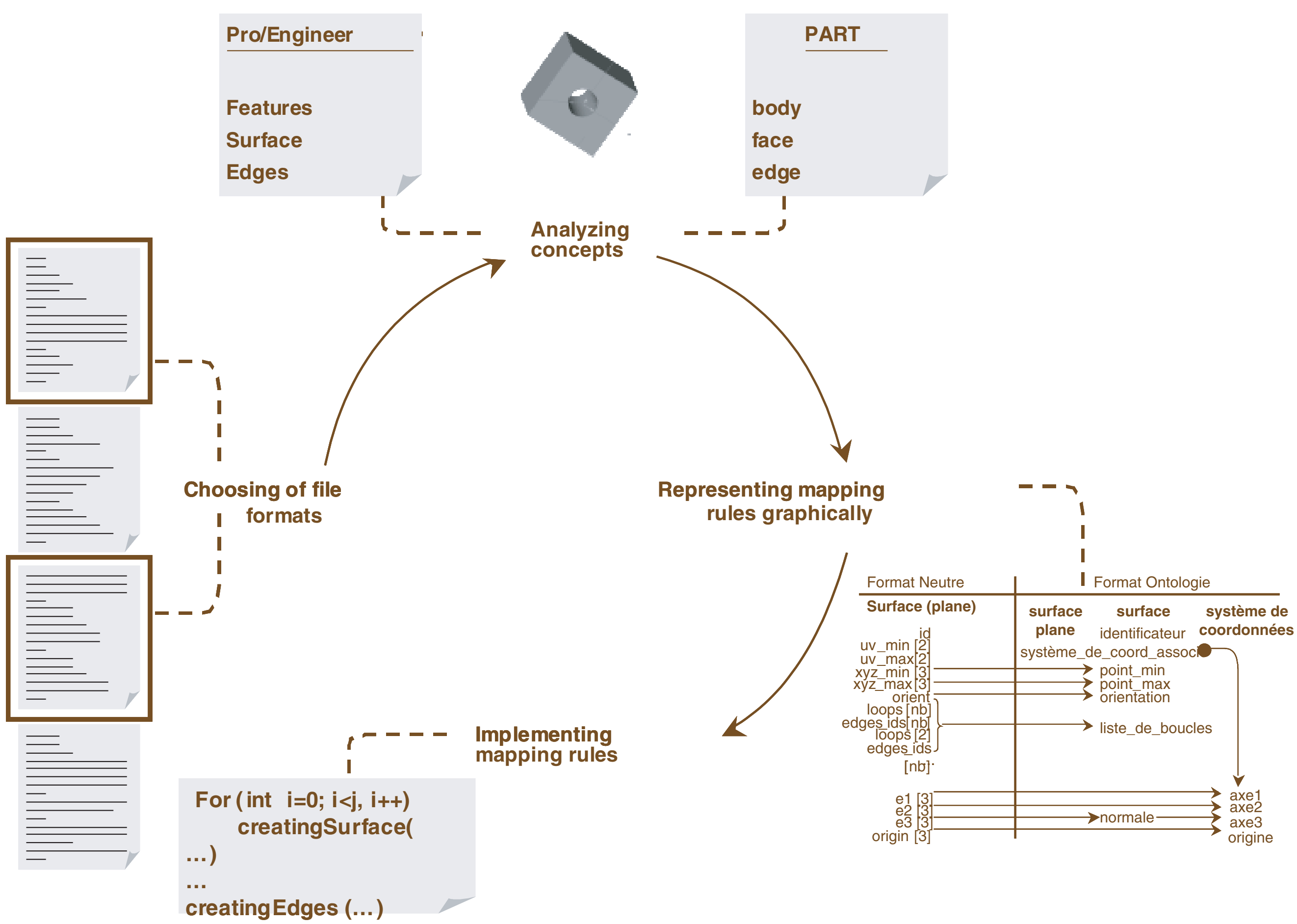

Figure 10. Methodology for the elaboration of the validation prototype.

experts, and PART, which is used by CAPP experts. The methodology followed is described in Figure 10.

The existing export and import formats of Pro-Engineer and PART were first analyzed. Then, one format was selected for each of them: a proprietary format for Pro-Engineer, called Neutral File Format, and ACIS format for PART. Once the formats have been chosen, the representation of

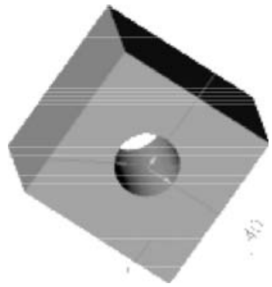

Figure 11. Data declaration in Pro-Engineer and PART files for a simple artifact.

different artifacts in the two formats was analyzed to extract all the important concepts represented in each file in order to correlate them with the domain ontology entities. Two algorithms were then defined: one to translate a file generated by CAD software into a set of instances of the feature ontology and one to translate this generated file into a file that can be interpreted and processed by a CAPP software. The inputs to the first algorithm are the file containing the entire description of the ontology, which is expressed in KIF, and the file generated by the CAD software (Pro/Engineer in this case), which represents the geometry and topology of the part that has to be manufactured. The inputs to the second algorithm are the file containing the entire description of the ontology, which is the common ontology expressed in KIF, and the file generated by the first algorithm.

As stated earlier, the only assumption made during the elaboration of the ontology and the mapping rules was that only parts that do not have any assembly were considered; solving this problem for simple machining parts containing only features by itself is a difficult task. Taking into account more complex parts containing for example assemblies would imply to modify both the ontology and the mapping rules.

For a simple artifact such as a box with one hole (Figure 11), the file generated by Pro-Engineer is hierarchically structured: it contains the dimensions characterizing the artifact, the features used to build it, the surfaces determining the features, and the edges composing the surfaces. PART files are totally different: information is stored with no specific order, and data contained in such files relate to geometric and topologic information. This kind of file format does not explicitly provide information about features composing an artifact.

Using different instances of Pro-Engineer and PART files, a list of entities or concepts and their attributes was extracted in these files, such as: plane surface, cylindrical surface, straight curve, linear curve, edge, point, vertex, etc. Once this analysis is done, the mapping rules between a Pro-Engineer file and a file containing instances of the ontology and between this generated file and a PART file were elaborated. The objective of these rules is to identify in the domain ontology the entities that are equivalent to the concepts identified in Pro-Engineer and PART files. These mapping rules were first expressed graphically. Figure 12 shows the graphical representation of one such mapping rule, which shows the correspondence between a plane surface expressed in a neutral file generated by Pro-Engineer and the equivalent concepts in the ontology.

Once this step is finished, two sets of mapping rules are obtained. These mapping rules are expressed graphically. The next step consists in implementing these rules in order to be able to translate a CAD file into a CAPP file via the ontology. As has been previously stated, the method involves starting from a file generated by Pro/Engineer, applying a first set of mapping rules in order to generate a neutral file, and applying the second set of mapping rules on this neutral file in order to obtain a file interpretable by PART. A description of the mapping algorithm is shown in Figure 13.

Starting from a CAD file generated by Pro/Engineer, all the features are created. For each feature, one has to extract from the ontology all the attributes that have been identified for a feature (for example the list of surfaces, the list of dimensions, etc.). For each of these attributes one searches, still in the ontology, the nature of the attribute, which can be either simple (i.e., integer, string, boolean) or complex (i.e., the attribute is composed of sub-attributes). If the attribute is a simple one, one extracts in the initial file the associated value and one adds a new instance in the neutral file. If the attribute is more complex, one considers each sub-attribute until all concepts appearing in the initial file have been instantiated. The advantage of this algorithm is that if one decides to change the attributes of one of the concepts of the ontology –for example if one deletes one attribute of the concept feature – the algorithm will not have to be changed because the number of attributes of a concept is calculated each time the algorithm is running.

The methodology and the prototype were tested with different examples. For the initial prototype only simple parts were considered (see Figure 14), with great success. The authors plan to extend this work for complex artifacts (e.g., assemblies).

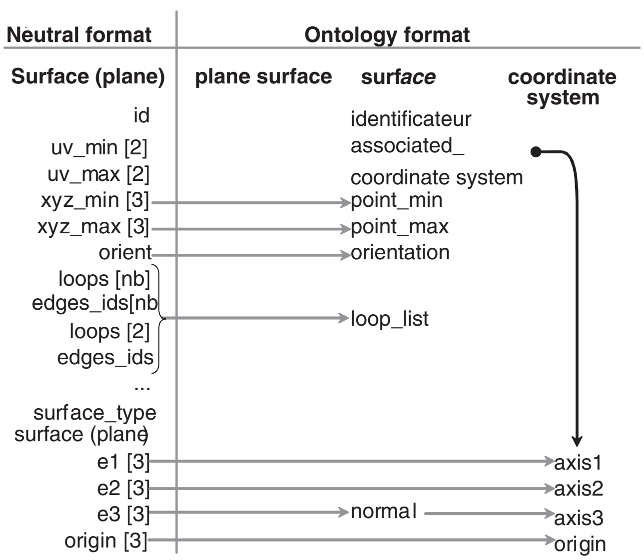

Figure 12. Correspondence for plane surface between a neutral file and the ontology.

9. Conclusions

In this study, an ontological approach to integrating computer-aided design (CAD) and computer-aided process planning (CAPP) has been described. Two commercial software applications were used to demonstrate the approach. The approach involved the development of a shared ontology and domain specific ontologies in the Knowledge Interchange Format (KIF) language. Domain specific ontologies – which were feature-based – were developed after a detailed analysis of the CAD and the CAPP software. Mapping between the domain ontologies and the shared ontology was achieved by several mapping rules. The approach was validated by using a variety of parts.

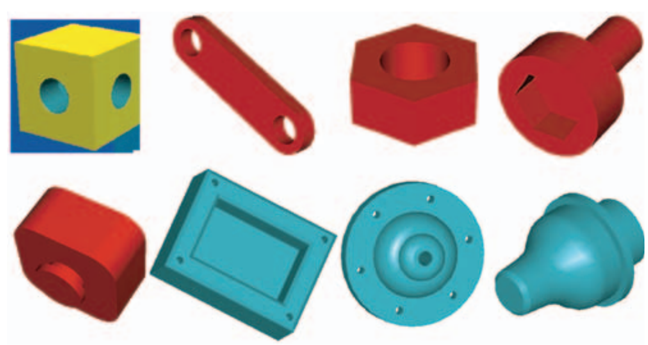

Figure 14. Some part examples.

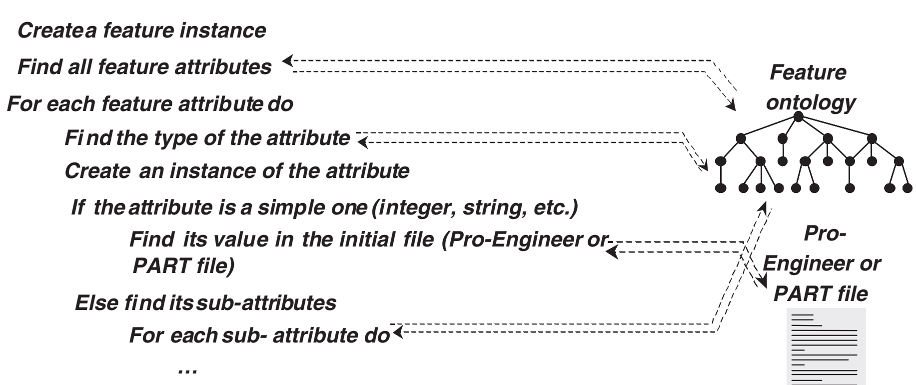

Figure 13. General algorithm for data exchange.

References

1. Eynard, B., Liénard, S., Charles, S. and Odinot, A. (2005). Web-based Collaborative Engineering Support System: Applications in Mechanical Design and Structural Analysis, *CERA Journal*, **13**(2): 145–153.
2. Jia, H.Z., Fuh, J.Y.H., Nee, A.Y.C. and Zhang, Y.F. (2002). Web-based Multi-functional Scheduling System for a Distributed Manufacturing Environment, *CERA Journal*, **10**(1): 27–39.
3. Han, S., Choi, Y., Yoo, S. and Park, N. (2002). Collaborative Engineering Design Based on an Intelligent STEP Database, *CERA Journal*, **10**(3): 239–249.
4. Huang, G.Q. and Mak, K.L. (2002). Agent-based Collaboration Between Distributed Web Applications: Case Study on "Collaborative Design for X" Using CyberCO, *CERA Journal*, **10**(4): 279–290.
5. Sriram, R.D. (2002). *Distributed and Integrated Collaborative Engineering Design*, Sarven Publishers.
6. Baker, B.A., Fish, R.D. and Cohen, E. (2000). Using a Multiple Concurrent Design Views Interface to Enhance Design Complexity Management, *ASME Design Engineering Technical Conferences (DETC)*, Baltimore, Maryland, USA.
7. Feng, S.C. and Song, E.Y. (2002). Preliminary Design and Manufacturing Planning Integration Using Intelligent Agents, In: *Proceedings of the Seventh International Conference on CSCW in Design*, pp. 270–275, Rio de Janeiro, Brazil.
8. Jan de Kraker, K. (1998). *Feature Conversion for Concurrent Enginnering, in Computer Science*, Delft University of Technology: Delft, The Netherlands.
9. Dym, C. (1994). *Engineering Design: A Synthesis of Views*, Cambridge, England: edition Cambridge University Press.
10. Fenves, S.J. (2001). *A Core Product Model for Representing Design Information*, National Institute of Standards and Technologies (NIST): Gaithersburg, MD, USA.
11. Algeo, M.E.A., Feng, S.C. and Ray, S.R. (1994). A State-of-the-Art Survey on Product Design and Process Planning Integration Mechanisms, NIST Interagency Report 5548, National Institute of Standards and Technology (NIST).
12. Ball, M., Baras, J., Lin, E., Minis, I., Nau, D. and Karne, R. (1998). Integrated Product and Process Design Tool for Microwave Modules, In: *Integrating Product Design and Production: Designing for Time-to-Market*, NSF Design and Manufacturing Grantees Conference, pp. 655–656. January 5–8, 1998, Monterrey, Mexico.
13. Han, J.H. and Requicha, A.A.G. (1995). Integration of Feature Based Design and Feature Recognition, *Proceedings of ASME 15th International Conference in Engineering Conference*, pp. 393–403, Boston, MA, USA.
14. Kramer, T.R. (1987). *Process Plan Expression, Generation, and Enhancement for the Vertical Workstation Milling Machine in the Automated Manufacturing Research Facility at the National Bureau of Standards*, National Bureau of Standards: Gaithersburg, MD, USA, NBSIR 87-3678, p. 56.
15. Regli, W.C. and Pratt, M. (1996). What are Feature Interactions *Proceedings of the 1996 ASME Design Engineering Technical Conference and Computers in Engineering Conference*, Irvine, California, USA.
16. Houten, F.J.A.M.v., Erve, A.H.v.t. and Kals, H.J.J. (1989). PART a Feature Based CAPP System, *21st CIRP International Seminar on Manufacturing Systems*, pp. 179–191, Stockholm, Sweden.
17. Feng, S.C. and Song, E.Y. (2000). Information Modeling on Conceptual Design Integrated with Process Planning, In: *Proceedings of Symposia for Design For Manufacturability*, The 2000 International Mechanical Engineering Congress and Exposition (IMECE), 2000, pp. 123–130, Orlando, FL, USA.
18. Han, J.-h., Pratt, M. and Regli, W.C. (2000). Manufacturing Feature Recognition from Solid Models: A Status Report, *Transactions on Robotics and Automation*, **16**(6): 782–796.
19. Han, J.H. (1996). *Survey of Feature Research*, Department of Computer Science and Institute for Robotics and Intelligent Systems, University of Southern California, Los Angeles, CA, USA.
20. De Martino, T., Falcidieno, B. and Hassinger, S. (1998). Design and Engineering Process Integration through a Multiple View Intermediate Modeller in a Distributed Object-oriented System Environment, *Computer-Aided Design*, **30**(6): 437–452.
21. De Martino, T., Falcidieno, B., Giannini, F., Hassinger, S. and Ovtcharova, J. (1994). Feature-based Modelling by Integrating Design and Recognition Approaches, *Computer-Aided Design*, **26**(8): 646–653.
22. Junhwan, K. and Soonhung, H. (2004). Manipulating Geometry in a STEP DB from Commercial CAD Systems, *CERA Journal*, **12**(1): 49–57.
23. Stephen, C.F.C., Tharam, D. and Vincent T.Y.N. (2003). Exchanging STEP Data through XML-based Mediators, *CERA Journal*, **11**(1): 55–64.
24. Balakrishna, A., Suresh Babu, R., Nageswara Rao, D., Ranga Raju, D. and Kolli, S. (2006). Integration of CAD/CAM/CAE in Product Development System Using STEP/XML, *CERA Journal*, **14**(2): 121–128.
25. Dereli, T. and Filiz, H. (2002) A Note on the Use of STEP for Interfacing Design to Process Planning, *Computer-Aided Design*, **34**: 1075–1085.
26. Bidarra, R. and Bronsvoort, W.F. (2000). Semantic Feature Modelling, *Computer-Aided Design*, **32**: 201–225.
27. Bronsvoort, W.F., Bidarra, R. and Noort, A. (2001). Semantic and Multiple-view Feature Modelling: Towards More Meaningful Product Modelling, Geometric Modelling, In: Kimura, F. (ed.), *Theoretical and Computational Basis towards Advanced CAD Application*, p. 384, Boston: Kluwer Academic Publishers.
28. Mäntylä, M., Nau, D. and Shah, J.J. (1996). Challenges in Feature-Based Manufacturing Research, *Communication of the ACM*, **39**(2): 77–85.
29. Pratt, M. and Srinivasan, V. (2003). *Towards a Neutral Specification of Geometric Features*, International Journal of Computer Applications in Technology 2005, **23**(2–4): 203–218
30. Salomons, O.W., Houten, F.J.A.M.v. and Kals, H.J.J. (1993). Review of Research in Feature Based Design, *Journal of Manufacturing Systems*, **12**(2): 113–132.
31. Shah, J.J. and Mäntylä, M. (1995). *Parametric and Feature-based CAD/CAM*, ed. John Wiley & Sons, Inc., New York, USA, p. 640.
32. Shah, J.J. and Rogers, M.T. (1988). Feature Based Modeling Shell: Design and Implementation, In: *Proceedings of the ASME Conference on Computers in Engineering*, pp. 343–354. Montreal, Quebec, Canada.
33. Collet, C., Huhns, M.N. and Shen, W. (1991). Resource Integration Using a Large Knowledge Base in Carnot, *IEEE Computer*, **24**: 55–62.
34. Hayes, E.E. and Menzel, C.P. (2001). A Semantics for Knowledge Interchange Format, *Working Notes of the*

IJCAI-2001 Workshop on the IEEE Standard Upper Ontology, Seattle, Washington, USA.
35. Genesereth, M. and Fikes, R. (2001). Knowledge Interchange Format, *Version 3.0 Reference Manual*, Computer Science Department, Stanford University, Stanford, CA, USA.
36. Genesereth, M.R. (2000). Knowledge Interchange Format, American National Standard (dpANS).
37. Sowa, J.F. (2000). Glossary.
38. Cutting-Decelle, A.F., Young, R.I.M., Anumba, C.J., Baldwin, A.N. and Bouchlaghem, N.M. (2003). The Application of PSL to Product Design Across Construction and Manufacturing, *CERA Journal*, **11**(1): 65–75.
39. Sudarsan, R., Han, Y., Feng, S.C., Roy, U., Wang, F., Sriram, R.D. and Lyons, K.W. (2003). *Object-Oriented Representation of Electro-Mechanical Assemblies Using UML*, National Institute of Standards and Technologies (NIST), pp. 228–234, Gaithersburg, MD, USA.
40. ISO10303-48 (1992). STEP Product Data Representation and Exchange, Integrated Generic Resources: Form Feature, International Organisation for Standardization, Subcommittee 4, Part 48, NIST.
41. Dartigues, C. (2003). Product Data Exchange in a Collaborative Environment, *Computer Science*, LIRIS, University of Lyon I, Lyon, France, p. 179.

C. Dartigues

Christel Dartigues is currently an associate professor in the Technological Institute of the University of Nice Sophia-Antipolis, in France, in the I3S laboratory. She obtained her PhD in computer sciences in December 2003 at the University of Lyon (France). She also spent 6 months as a guest researcher at the National Institute of Standards and Technology (NIST), at Gaithersburg (USA). Her research interest is in the area of collaborative design, ontologies and e-Learning.

P. Ghodous

Parisa Ghodous is currently professor in computer science department of University of Lyon I. She is head of collaborative modeling team of LIRIS (Lyon Research Center for Images and Intelligent Information Systems), UMR 5205.

Her research expertise is in the following areas: Collaborative Modeling, Interoperability, Distributed Data Base, Product data exchange and Standards. She is in editorial boards of 6 journals and in the committees of many relevant international associations, such as concurrent engineering, ISPE, ECPPM, Interoperability...

D. Pallez

Denis Pallez is currently an associate professor in the department of computer sciences at the University of Lyon, France, in the LIRIS laboratory (Lyon Research Center for Images and Intelligent Information Systems). He obtained his PhD in computer sciences in January 2000 at the University of Metz (France). He also spent 4 months as an invited researcher at the institute of technology at Montreal (ETS). His research interest is in the area of conceptual and functional design, shape synthesis, collaborative modeling and more recently, Interactive Evolutionary Computation.

M. Gruninger

Michael Gruninger is currently an assistant professor of Mechanical and Industrial Engineering, University of Toronto, Canada. Previously, he was a research scientist in the Institute for Systems Research at the University of Maryland College Park and also a guest researcher at the National Institute for Standards and Technology (NIST). His current research focuses on the design and formal characterization of ontologies and their application to problems in manufacturing and enterprise engineering. He was a project leader for the Process Specification Language project at NIST. He is also the project leader for ISO 18629 (Process Specification Language) within the International Standards Organization (ISO), and project editor for the Knowledge Interchange Format, also within ISO. Michael received his PhD and MSc in Computer Science at the University of Toronto and his BSc in Computer Science at the University of Alberta.

R. Sriram

Ram D. Sriram is currently leading the Design and Process group in the Manufacturing Systems Integration Division at the National Institute of Standards and Technology, where he conducts research on standards for inter-operability of computer-aided design systems and on healthcare informatics. He is also the manager for the 'Manufacturing Metrology and Standards for the Healthcare Enterprise Program' in the Manufacturing Engineering Laboratory. Prior to that he was on the engineering faculty (1986–1994) at the Massachusetts Institute of Technology (MIT) and was instrumental in setting up the Intelligent Engineering Systems Laboratory. At MIT, Sriram initiated the MIT-DICE project, which was one of the pioneering projects in collaborative engineering. Sriram has co-authored or authored nearly 200 papers, books, and reports in computer-aided engineering, including several books. Sriram was a founding co-editor of the *International Journal for AI in Engineering*. In 1989, he was awarded a Presidential Young Investigators Award by the National Science Foundation, USA. Sriram has a BS from IIT, Madras, India, and an MS and a PhD from Carnegie Mellon University, Pittsburgh, USA.